In re Patent Application of:
CAVALLI
Serial No. **09/826,427**
Filing Date: **April 3, 2001**
_____/

In the Claims:

1. (Currently amended) A method for processing an interrupt signal using a microprocessor comprising a central processing unit (CPU), an interrupt controller providing an interrupt request signal to the CPU when the interrupt signal is received by the microprocessor, registers having a context stored therein corresponding to a program being executed by the CPU, and a stack for storing the context while an interrupt is being executed, the method comprising:

detecting a receipt of the interrupt request signal by the CPU from the interrupt controller;

storing the context from the registers to the stack;

verifying that the interrupt request signal is still provided to the CPU from the interrupt controller after storing the context to the stack;

sending an interrupt acknowledge signal and reading and executing a first instruction of an interrupt subroutine using the CPU if the interrupt request signal is still provided to the CPU; and

restoring the stored context from the stack to the registers and returning the CPU to an initial state if the interrupt request signal is not still provided to the CPU.

2. (Original) The method according to Claim 1 wherein reading and executing the first instruction of the interrupt subroutine comprises reading a data element from an address determined based upon an interrupt vector provided by the

interrupt controller, the data element comprising a read address of the first instruction of the interrupt subroutine.

3. (Original) The method according to Claim 1 wherein returning the CPU to the initial state comprises reading and executing an instruction of the program being executed by the CPU when the interrupt request signal was detected.

4. (Original) The method according to Claim 1 wherein returning the CPU to the initial state comprises processing a new interrupt if a new interrupt request signal is provided by the interrupt controller.

5. (Original) The method according to Claim 1 wherein restoring the stored context from the stack comprises restoring contents of a program counter register.

6. (Original) The method according to Claim 5 wherein returning the CPU to the initial state comprises providing the restored contents of the program counter register to an address bus.

7. (Currently amended) A method for processing an interrupt request using a microprocessor executing a program, the method comprising:

detecting the interrupt request;

storing contextual data of the program;

~~sending an interrupt acknowledge signal;~~

verifying that the interrupt request is still present after storing the contextual data;

resuming execution of the program if the presence of the interrupt request is not verified; and

sending an interrupt acknowledge signal and switching to an interrupt subroutine if the presence of the interrupt request is verified.

8. (Original) The method according to Claim 7 wherein storing contextual data comprises storing data located in registers of the microprocessor in a random-access memory.

9. (Original) The method according to Claim 8 wherein the data stored in the random-access memory are stored into respective original registers before resuming execution of the program.

10. (Original) The method according to Claim 9 wherein resuming execution of the program is postponed if a new interrupt is detected after the presence of the interrupt request is not verified; and further comprising processing the new interrupt request.

11. (Currently amended) A microprocessor comprising:

registers having a context stored therein;

a stack for storing the context;

an interrupt controller for providing an interrupt request and an interrupt vector when an interrupt signal is

applied to the microprocessor; and

a CPU for receiving the interrupt request and the interrupt vector and, upon detection of the interrupt request

storing the context in said stack,

verifying that the interrupt request is still present after storing the context,

sending an interrupt acknowledge signal and reading and executing a first instruction of an interrupt subroutine if the presence of the interrupt request is verified, and

restoring the stored context from said stack and returning the microprocessor to an initial state if the presence of the interrupt request is not verified.

12. (Original) The microprocessor according to Claim 11 wherein said CPU reads a data element from an address determined based upon the interrupt vector, the data element comprising a read address of the first instruction of the interrupt subroutine.

13. (Original) The microprocessor according to Claim 11 further comprising a program counter register; and wherein said CPU restores contents of said program counter register during restoring of the stored context.

14. (Original) The microprocessor according to Claim 13 wherein during returning the microprocessor to the initial state said CPU reads and executes an instruction corresponding to

an address in said program counter register.

15. (Currently amended) ~~A~~ The microprocessor according to Claim 11 wherein during returning the microprocessor to an initial state said CPU processes a new interrupt if a new interrupt request is provided by said interrupt controller.

16. (Currently amended) ~~A~~ The microprocessor according to Claim 11 further comprising:

a 16-bit program counter register comprising two 8-bit registers; and

a 16-bit stack pointer register comprising two 8-bit registers.

17. (Currently amended) A microprocessor comprising:

at least one memory; and

a central processing unit (CPU) for

detecting an interrupt request during execution of a program,

storing contextual data of the program being executed in said at least one memory,

sending an interrupt acknowledge signal and switching to an interrupt subroutine if the interrupt request is still present after storing the contextual data, and

resuming execution of the program if the interrupt request is not still present after storing the contextual data.

18. (Currently amended) The microprocessor according to Claim 17 ~~further comprising~~ wherein said at least one memory comprises a random-access memory and registers for storing data elements; and wherein said CPU stores the contextual data by storing the data elements in said random-access memory.

19. (Original) The microprocessor according to Claim 18 wherein said CPU stores the data elements stored in the random-access memory in respective original registers thereof before resuming execution of the program.

20. (Currently amended) The microprocessor according to Claim 17 wherein said CPU processes a new interrupt request if the new interrupt request is present ~~after~~ and the interrupt request is not present ~~and~~ before resuming execution of the program.